ตอน 9

แนะนำ AutoCAD เวอร์ชัน 10

สุรพล พฤกษ์ไพบูลย์

การแข่งขันในวงการ CAD ชักเริ่มจะออกรสชาติหลังจากที่ออโต้เดสค์บริษัทผู้ผลิตออโต้แคดซึ่งครองความเป็นหนึ่งในตลาดด้านนี้มาเป็นเวลานานแสนนาน ยังไงก็ไม่ยอมเสียส่วนแบ่งตลาดทางด้าน CAD บนไมโครคอมพิวเตอร์ให้กับใคร(หลังจากที่แน่ใจตัวเองว่าเป็นหนึ่งเสมอ) จึงตัดสินใจพัฒนาตัวเองครั้งใหญ่เพิ่มขีดความสามารถของตัวโปรแกรมโดยจัดการเก็บข้อมูลของตัวเองเสียใหม่ ให้เป็นระบบสามมิติสมบูรณ์แบบเป็นเวอร์ชัน 10 ทำให้เราสามารถเรียกออโต้แคดว่าเป็นโปรแกรม CAD ได้อย่างเต็มปากเต็มคำ ซึ่งถ้ามองสภาพการแข่งขันของตลาดด้านนี้แล้วเราจะพบว่าออโต้เดสค์อยู่ในฐานะลำบากที่จะรักษาความเป็นหนึ่งให้ยืนยงคงอยู่ได้ เนื่องจากในปัจจุบันเครื่องคอมพิวเตอร์ระบบไมโครฯ ได้พัฒนาตัวเองจนมีขีดความสามารถอย่างล้นเหลือ ไหนจะมีโปรแกรมหน้าใหม่อย่างดีไซน์แคด 3D ที่มีความสามารถทางสามมิติที่ออโต้แคดไม่มี ไหนจะต้องขึ้นไปแข่งกับโปรแกรมบนระบบยูนิกซ์เพราะว่าเครื่อง 386 มีความสามารถที่จะใช้ระบบยูนิกซ์ได้ ก็...บนระบบยูนิกซ์น่ะมีทั้งโปรแกรมคอมพิวเตอร์วิชัน ของบริษัทคอมพิวเตอร์วิชัน (ซึ่งตอนนี้ได้กลายเป็นส่วนหนึ่งของ PRIME ไปแล้ว) และโปรแกรม ARRIS ของบริษัทซิกม่าดีไซน์ (คนละเจ้ากับที่ผลิตดิสเพลย์การ์ด) เป็นหนึ่งครองตลาด CAD บนระบบยูนิกซ์มานานช้านาน ARRIS เป็นโปรแกรมเจ้าเก่าที่มีแทบทุกอย่างที่โปรแกรม CAD ในอุดมคติควรจะมี (เช่น การวิเคราะห์เนื้อที่ใช้สอย ซึ่ง ARRIS เรียกว่าสเปซดีไซน์ ดังนั้นถ้าออโต้แคดยังคงเป็นอยู่เช่นเวอร์ชัน 9 ละก้อ รับรองได้ว่าในไม่

ช้าจะต้องหน้าแตกสูญเสียตลาดด้านนี้ไปอย่างไม่ต้องสงสัย และยังจะไปทำเท่ห์บนระบบยูนิกซ์ไม่ได้อีกด้วย ไม่ใช่ว่าออโต้แคดจะไม่มีดี ก็ระบบ SNAP (เช่น endpoint, insert, intersect, mid-point) บนออโต้แคดที่ช่วยการเขียนให้ง่ายขึ้นไงล่ะครับ ที่โปรแกรมอื่น ๆ อีกหลายตัวยังสู้ไม่ได้... และตำนานการต่อสู้ก็ยังคงดำเนินต่อไปเรื่อย ๆ โดยทิ้งให้พวกเราเป็นสักขีพยานในการชิงชัยครั้งนี้

ระบบ UCS

อันที่จริง ออโต้แคด เวอร์ชัน 10 นี้ก็ยังคงรูปร่างภายนอกคือเมื่อเรียกโปรแกรมขึ้นมาและเข้าไปใน drawing editor แล้วมีลักษณะที่เหมือนกับรุ่นเก่าทุกประการ ยกเว้นเครื่องหมายประหลาดปรากฏอยู่ที่มุมซ้ายมือที่ออโต้แคด เรียกว่า UCSICON (มาจาก User Coordinate Icon) และนั่นคือสัญลักษณ์อันสำคัญสำหรับออโต้แคดรุ่นนี้ User Coordinate นี้เป็นความคิดของผู้ผลิตในการตอบปัญหาการเขียนรูปสามมิติในออโต้แคด ยกตัวอย่างเช่น ถ้าท่านใช้โปรแกรมรุ่นเก่าโดยใช้คำสั่ง VPOINT เปลี่ยนมุมมองเป็นรูปตั้งแล้ว จากนั้นท่านจะไม่สามารถที่จะทำอะไรกับรูปได้เลย แต่สำหรับโปรแกรมรุ่น 10 นี้ เมื่อท่านใช้ VPOINT เปลี่ยนมุมมองเป็นรูปตั้งแล้ว จากนั้นใช้คำสั่ง UCS แล้วเลือก View เป็นความหมายว่าให้เซต UCS ให้เป็นไปตามรูปที่ปรากฏที่หน้าจอ ท่านจะได้เห็นสัญลักษณ์ UCSICON ที่มุมล่างซ้ายเปลี่ยนแปลงไป เพื่อที่จะให้ท่านสามารถเลือกจุดมองโดยการ ZOOM หรือแม้แต่เขียนเส้นตามแกน Z ที่เป็นด้านตั้งได้ แต่การใช้ UCS นี้นำความยุ่งยากมาให้เราหลายประการ ประการแรก ไม่มีข้อมูลเกี่ยวกับแกน Z ขึ้นบนจอที่ตำแหน่งโคออร์ดิเนต ดังนั้นเราจึงไม่สามารถทราบได้ทันทีโดยไม่ได้เช็คว่าเราเขียนอยู่ที่ตำแหน่งใดของแกน Z และประการที่สอง เนื่องจากจะด้วยเหตุที่ว่ายังไม่พร้อมที่จะแสดงข้อมูลทางสามมิติหรืออะไรก็ตาม เครื่องหมาย UCS จะมีบอกเฉพาะแกน X และ Y เท่านั้น นั่นหมายความว่าเมื่อต้องการที่จะเขียนวัตถุบนแกน Z เราจะต้องแลกแกน Z กับแกน X หรือ Y แกนใดแกนหนึ่งเสมอ และการเปลี่ยนแกนกันนี้ค่าของโคออร์ดิเนตจะถูกสลับเปลี่ยนแปลงไประหว่าง X กับ Z หรือ Y กับ Z อย่างใดอย่างหนึ่งขึ้นอยู่กับมุมมองในขณะนั้น ๆ ว่าเป็นอย่างไร และนั่นรวมแล้วเป็นความสับสนที่ออโต้แคดมอบให้แก่เราในการเขียนรูปสามมิติ ถ้าเทียบกับโปรแกรมอื่น เอ้า! ไม่ต้องไปถึงเครื่องระดับมินิหรือเวอร์กสเตชันด้วย ในเครื่องไมโครฯด้วยกันโปรแกรมอย่าง Design Board Professional อธิบายการเขียนรูปสามมิติได้ชัดเจนดีกว่า โดยที่ไม่ต้องไปสลับเปลี่ยนแกน Z ให้ยุ่งยากเลย

ที่กล่าวมาทั้งหมดข้างต้นนี้ ไม่ใช่ว่าระบบ UCS ของออโต้แคดจะไม่ดี ข้อดีของ UCS ก็คือช่วยให้เราสามารถที่จะเขียนรูปต่าง ๆ ได้อย่างอิสระทุก ๆ ระนาบ โดยไม่จำกัดว่าแนวแกนต่าง ๆ จะตั้งฉากกันหรือไม่ เช่น แกน Z ก็ไม่จำเป็นจะต้องตั้งฉากกับแกน X และ Y ดังนั้นเราจึงสามารถที่จะใช้งานออโต้แคดได้อย่างคล่องตัวมาก การเปลี่ยนค่าและตำแหน่งของ UCS ทำให้เราสามารถเขียนหรือใช้คำสั่งที่ปกติจะกระทำได้เฉพาะบนระนาบพื้นราบ X-Y เท่านั้นเช่น HATCH บนพื้นที่เอียง หรือการทำลวดลายกระเบื้องบนหลังคาเอียงได้โดยไม่ยากลำบากเลย การเปลี่ยนแปลง UCS ให้เป็นไปตามพื้นผิวเอียงกระทำได้โดย

Command: UCS
Origin/ZAxis/3point/Entity/View/X/Y/Z/Prev/Restore/Save/Del/?<World>:3
Origin point<0,0,0>: <เลือกจุดที่มุมใดมุมหนึ่งของพื้นระนาบ>
Point on positive portion of the x-axis: <จากนั้นเลือกจุดที่อีกมุมหนึ่งที่อยู่ตามแนวแกน X จุดนี้จะอยู่ทางขวามือของจุดแรกถ้าเขียนรูปปกติ และอยู่ทางซ้ายถ้าเขียนกลับหัว>
Point on positive portion of the y-axis: <และอีกมุมหนึ่ง สำหรับแกน y>

หลังจากเลือกมุมของพื้นเอียง 3 จุดแล้ว จะเห็น UCSICON เปลี่ยนไปเอียงตามพื้นผิวที่ต้องการ

FLATLAND

ถ้าท่านยังไม่คุ้นเคยกับการเขียน

รูปในสามมิติที่จะต้องระมัดระวังเกี่ยวกับ Z-Coordinate ตลอดเวลาแล้วละก้อ ออโต้แคดอนุญาตให้เราเขียนรูปตามแบบเดิมโดยที่ค่าของ Z เป็นค่าคงที่ (ตามที่ได้เซตเอาไว้โดยใช้คำสั่ง ELEV) โดยการเซตค่าของตัวแปรระบบ (System Variable) ที่มีชื่อว่า FLATLAND ให้มีค่าเท่ากับ 1 ซึ่งก็จะทำให้ object snap ต่าง ๆ ไม่ "วิ่ง" ไปหาจุดตามตำแหน่ง Z จริงของออปเจคนั้น ๆ เราสามารถใช้ออโต้แคดรุ่น 10 นี้เหมือนกับรุ่นเก่าทุกประการไม่มีข้อเสียใด ๆ ทั้งสิ้น ยกเว้นอยู่ประการเดียวคือ ออโต้แคดบอกว่าจะยกเลิกตัวแปรที่ชื่อ FLATLAND นี้ในออโต้แคดรุ่นต่อไปข้างหน้า เพราะฉะนั้นก็ระวังเอาไว้หน่อย ข้อแนะนำของผมก็คือ ควรที่จะทำความคุ้นเคยกับระบบทางสามมิติต่าง ๆ เอาไว้เสียแต่บัดนี้ครับ!

GR.LSP

เมื่อวันก่อนผมมีโอกาสไปร่วมสัมมนาเรื่องออโต้แคด เวอร์ชัน 10 นี้ที่มหาวิทยาลัยศิลปากร โดยมีฝรั่งมาเป็นวิทยากร ซึ่งเป็นบุคคลที่มีความสามารถมากมาจากบริษัทออโต้เดสก์ที่ดูเหมือนจะมีสาขาอยู่ที่ฮ่องกง แต่เสียดายที่ผมจำชื่อไม่ได้เนื่องจากปกติก็เป็นคนที่มีความจำไม่ค่อยจะดี อีกทั้งวันนั้นก็เจ้ากรรม...เผอิญมีธุระทำให้ได้ไปเข้าฟังก็เมื่อตอนช่วงบ่ายเสียแล้ว อย่างไรก็ตามเมื่อมีผู้เยี่ยมยุทธในแนวทางนี้เราก็จะต้องไปดูกันเสียหน่อยใช่ไหมครับ และก็ได้ความสะใจพอสมควรสำหรับความคิดต่าง ๆ ที่จะนำมาพัฒนาการเขียนโปรแกรมออโต้ลิสป์ แต่คิดว่าเขาคงจะไม่ใช่คนที่มีพื้นฐานทางสถาปัตยกรรม ดังนั้นเราจึงไม่มีโอกาสเห็นโปรแกรมการช่วยเขียนทางด้านนี้เท่าไรนัก มีสิ่ง

หนึ่งที่ช่วยในเรื่องของ CAD นี้มากนั่นคือ ฮาร์ดแวร์ และความรวดเร็วในการแสดงผลและการคำนวณต่าง ๆ ซึ่งขึ้นอยู่กับตัวนี้มาก ในวันนั้นดูเหมือนเขาจะใช้เครื่อง 386 (ขี้ห้อ...อย่าบอกดีกว่านะครับ เพราะไม่ได้โฆษณา) มีหน่วยความจำหลักขนาด 2 เมกะไบต์พร้อมทั้ง Cached memory ส่วนดิสเพลย์การ์ดเป็น Vedio7 VGA ที่แสดงผลได้รวดเร็วมาก (การ์ด VGA แต่ละตัวมีความเร็วไม่เท่ากัน)

นอกจากนี้ก็มีโปรแกรมหนึ่งเป็นโปรแกรมการเขียนบอกกริด เสา และแนวระยะระหว่างช่วงเสา, เส้นบอกระยะ ฯลฯ ซึ่งที่จริงผมได้เขียนโปรแกรมทำนองนี้ขึ้นใช้งานที่บริษัทไว้นานแล้ว แต่เป็นลักษณะที่แตกต่างกันออกไป พอกลับมาถึงเพื่อนบอกว่าไหนลองทำอย่างที่ฝรั่งทำให้ดูวันนั้นหน่อย ครับ! บางทีก็เป็นอย่างนี้ พวกเรากันเองทำขึ้นมาบอกว่าเป็นเรื่องธรรมดา ๆ ทีฝรั่งทำมั่งก็ยกนิ้วบอกว่าเยี่ยม ๆ เพื่อนบอกว่าไม่ใช่อย่างนั้น เพียงแต่อยากจะดูว่าจะแตกต่างกันอย่างไรเท่านั้น ไอ้ผมมันก็คนใจอ่อนเลยลองเขียนโปรแกรมให้ดูออกมาคล้าย ๆ กับที่เห็นวันนั้น (อย่างไม่กลัวโดนฟ้องว่า look and feel เหมือนกัน อย่างที่ดีเบสฟ้องฟอกซ์เบส) ผลออกมาก็ทำงานได้เหมือนกัน จะต่างกันก็ตรงความคล่องตัวบางอย่าง โดยที่ของเก่าที่ผมเขียนเอาไว้แล้วสามารถที่จะใส่ระยะต่าง ๆ เข้าไปทีเดียวบรรทัดเดียวได้เลย ซึ่งแตกต่างกับของที่เขียนขึ้นใหม่นี้ แต่ผมจะขอนำแต่โปรแกรมที่เขียนขึ้นใหม่นี้มาลงเท่านั้น เพราะอาจมีผู้อ่านบางท่านที่ไปสัมมนาวันนั้นเหมือนกันนึกถึงได้ และอยากจะได้โปรแกรมเอาไว้ช่วยในงานบ้าง

โปรแกรมนี้ชื่อว่า GR.LSP ในรูปที่ 1 โดยก่อนที่จะใช้โปรแกรมนี้ได้จะต้องมีโปรแกรมอื่นอีก ซึ่งเป็นโปรแกรมที่ได้ลงไว้ในไมโครฯ เล่มที่แล้ว (เล่มที่ 47) คือ Formlist แต่เพื่อความสมบูรณ์ของเนื้อหาจะได้ขอนำมาลงอีกครั้งสำหรับผู้ที่ไม่ได้ติดตามบทความนี้ต่อเนื่องกัน ตามที่ได้กล่าวไว้แล้วในบทก่อนว่าโปรแกรม Formlist นี้ เป็นโปรแกรมที่มีประโยชน์อันหนึ่งในการเตรียมลิสต์เอาไว้เพื่อที่จะนำเอาไปใช้ต่อไป เรียกว่าเป็นโปรแกรมแบบอรรถประโยชน์ (Utility) และโปรแกรมนี้ผมได้นำมาช่วยในการเขียนโปรแกรมอื่น ๆ อีกมาก แต่ครั้งนี้จะขอนำมาลงอีกเป็นครั้งสุดท้ายเท่านั้น ส่วนโอกาสต่อไปข้างหน้าถ้ามีการอ้างถึงโปรแกรมนี้อีกก็ขอได้โปรดย้อนกลับมาดูในไมโครฯ เล่มดังกล่าว หรือถ้าจะให้ดีขอให้ท่านบรรจุโปรแกรมนี้รวมเอาไว้ใน ACAD.LSP เพื่อที่จะได้มีไว้ใช้ตลอดไป

โปรแกรมที่ได้แสดงเอาไว้ในที่นี้จะประกอบด้วยสองส่วน คือ ส่วนแรกจะเป็นตัวอย่างโปรแกรม ACAD.LSP

รูปที่ 1 โปรแกรม GR.LSP

```
; GR.LSP
; Author:.............Surapon Preugpaibul
; First created date:.....................Mar 11,1989
; Last updated date:......................Mar 11,1989
; Draw grid lines and dimensions
; Defaults:
;            Layer: GRID
;            color: 10
;            Numeric precision: 2
;            Text in buble: 0.5
;                 color: 2
;            Text in dimension: 0.25
;                 color: bylayer
;
; Return the list after taking the last one out
;
(defun lastout(l)
  (reverse (cdr (reverse l)))
)

(defun C:GR (/ la na a0 b0 a b d ang a1 a2 n0)
  (graphscr)
  (pushvars '("cmdecho" "orthomode" "blipmode" "texteval" "dimtxt"
              "dimtix" "dimsoxd" "luprec"))
  (setqv '("cmdecho" 0 "blipmode" 1 "orthomode" 1 "luprec" 2
           "texteval" 1 "dimtxt" 0.25  "dimtix" 1 "dimsoxd" 1))
  (setq la (getvar"clayer")
        a0 (getpoint"\nFrom point: ") b0 (getpoint a0 "\nTo point: ")
        d (distance a0 b0) ang (angle a0 b0)
        a (formlist "\nGrid point: " a0) a (cons a0 a)
        b (mapcar '(lambda (x) (polar x ang d)) a)
        a1 (lastout a) a2 (cdr a)
  )
  (if (or(eql ang 0) (eql ang pi))
    (progn
    (setq s "VER" na (getstring"\nBegining character <A>: "))
    (if (= na "") (setq na "A") (setq na (strcase na)))
    )
    (progn
    (setq s "HOR" na (getint "\nBegining character <1>: "))
    (if (or(null na)(< na 1)) (setq na 1))
    )
  )
  (if (null(tblsearch "layer" "grid"))
    (command "layer" "m" "grid" "c" 10 "" "")
    (command "layer" "s" "grid" "")
  )
  (mapcar '(lambda (x y / xl)
          (setq xl (polar x (+ ang pi) 2.4))
          (command "linetype" "s" "center" ""
                   "line" x y ""
                   "linetype" "s" "continuous" ""
                   "insert" "bub" xl 1 1 (rtod ang)
                   "color" 2 "text" "m" xl 0.5 0 na
                   "color" "bylayer"
          )
          (if (numberp na) (setq na (1+ na)) (setq na (chr(1+ (ascii na))))
          ) a b)
  (mapcar '(lambda (x y) (command "diml" s x y x "")) a1 a2)
  (command"layer" "s" la "")
  (popvars)
)
```

รูปที่ 2 โปรแกรม BLOCKBUB.LSP

```
; Create block named BUB; for use with GR.LSP;
(apply '(lambda ()
  (command "circle" '(1.0 1.0) 0.75
           "line" '(1.75 1.0) '(3.75 1.0) ""
           "block" "BUB" '(1.0 1.0) "c" '(1.75 1.0) '(1.75 1.0) ""
           "wblock" "BUB" "="
   )
   )
   '()
)
```

ซึ่งได้รวบรวมโปรแกรมอรรถประโยชน์เอาไว้ เพื่อสำหรับโปรแกรมอื่น ๆ และส่วนที่สองจะเป็นโปรแกรม GR.LSP นอกจากนี้แล้วท่านจะต้องใช้โปรแกรม BLOCKBUB ในรูปที่ 2 ซึ่งให้ท่านเรียกใช้เพียงครั้งเดียว โดยการ (load "blockbub") ที่ command: ของออโต้แคดจะเป็นการเตรียมบล็อกสำหรับโปรแกรม GR อีกทีหนึ่ง โปรแกรมนี้จะแสดงให้ท่านเห็นว่าเราสามารถที่จะสร้างบล็อกขึ้นโดยที่ไม่ต้องออกแรงเขียนเลยแม้แต่เส้นเดียวได้อย่างไร

ส่วนที่สองจะเป็นตัวโปรแกรม GR ซึ่งใช้วาดแนวกริด เสา และระยะต่าง ๆ การใช้สัญลักษณ์ต่าง ๆ รวมทั้งขนาดตัวหนังสือ, ขนาดของเส้น (สี), รูปแบบต่าง ๆ ของตัวหนังสือ ฯลฯ ที่ใช้ในโปรแกรมนี้ มิได้หมายความว่าที่บริษัทผมใช้ตามแบบนี้ เพียงแต่ผมได้ให้เอาไว้กลาง ๆ อีกทั้งเพื่อความสะดวกในการแสดง เช่นบล็อกที่ใช้แทนลูกศร ผมก็ไม่ได้เปลี่ยนใหม่ยังคงใช้เป็นรูปลูกศรตามแบบของออโต้แคด หากท่านต้องการใช้เป็นรูปวงกลม ก็สามารถเปลี่ยนได้แต่ท่านจะต้องสร้างบล็อกรูปที่ต้องการลงในขนาด 1 ยูนิต แล้วใช้คำสั่ง BLOCK ในชื่อที่ต้องการ จากนั้นใช้คำสั่ง WBLOCK เพื่อเก็บบล็อกลงในดิสก์ เมื่อใช้งานจะต้องเซตชื่อในตัวแปรระบบที่มีชื่อว่า dimblock ให้เป็นชื่อบล็อกของเรา (ดูรายละเอียดใน AutoCAD Reference Manual หน้า 286) ส่วนค่าอื่น ๆ เช่น layer ที่ใช้, ขนาดตัวหนังสือก็สามารถเปลี่ยนแปลงได้โดย edit ในโปรแกรมเลย **ก่อนที่จะเริ่มโปรแกรม GR ขอให้เรียกโปรแกรม BLOCKBUB เสียก่อน**

ในที่นี้จะขออธิบายแต่เฉพาะตัวโปรแกรม GR เพราะว่าโปรแกรมอื่น ๆ ได้อธิบายเอาไว้แล้วในไมโครฯ เล่มที่ 47 ขอให้ดูทีละช่วงนะครับ บรรทัดแรก (graphscr) เป็นการเปลี่ยนหน้าจอให้เป็นจอกราฟิก ซึ่งถ้าเป็นกราฟิกอยู่แล้วฟังก์ชันนี้ก็ไม่ทำอะไร จากนั้นจะเป็นชุดการเก็บค่าของตัวแปรระบบเข้าไว้ (ใน _vars) โดยใช้ฟังก์ชัน pushvars (อยู่ใน ACAD.LSP ในรูปที่ 3) และต่อมาเป็นการเซตค่าของตัวแปรระบบนี้เป็นค่าต่าง ๆ โดยใช้ setqv (อยู่ใน ACAD.LSP เช่นเดียวกัน) บรรทัดต่อมาจะเป็นชุดของ setq ซึ่งตัวแรกคือ la จะเก็บชื่อของ layer ปัจจุบันเอาไว้ (เพื่อไว้เซตกลับคืนภายหลัง) และ na เป็นตัวอักษรที่ใช้เริ่มต้นกริด เช่น กริดตามแนว

รูปที่ 3 โปรแกรม ACAD.LSP

```
; Example of ACAD.LSP
; Author:.............Surapon Preugpaibul
;
(vmon)
;
; Change radian to degree
(defun rtod (a)
   (/ (* 180.0 a) pi)
```

รูปที่ 3 (ต่อ) โปรแกรม ACAD.LSP

```
)
; Change degree to radian
(defun dtor (a)
   (/ (* pi a) 180.0)
)

(defun formlist (s e / a)
  (cond ((= (type e) 'str) (setq a (getstring 't s)))
        ((= (type e) 'list) (setq a (getpoint e s) e a))
        ((= (type e) 'int) (setq a (getint s)))
        ((= (type e) 'real) (setq a (getreal s)))
        ((= (type e) 'sym)
          (setq a (getstring 't s))
          (if (/= a "") (setq a (read a)))
        )
        ((null e) (setq a (getpoint s)))
  )
  (if (listp a)
    (if a (cons a (formlist s e)))
    (if a
      (if (/= a "")
        (cons a (formlist s e))
      )
    )
  )
)
;
; "Setq" type of setvar system variable
(defun setqv (a)
  (if a (progn (setvar (car a) (cadr a)) (setqv (cddr a))))
)
;
; (eql <number1> <number2>)
; Test for equality of two real numbers
; This also should be in ACAD.LSP
;
(defun eql (a b)
  (cond ((= a b) 't)
        ((> a b) (if (< (- a b) 0.0000000000001) 't nil))
        ((> b a) (if (< (- b a) 0.0000000000001) 't nil))
        (t nil)
  )
)
;
; Save system variables in _VARS.  The read-only variables may be used.
(defun PUSHVARS (a)
   (setq _vars (append
                 (mapcar 'list
                    (mapcar 'strcase a)
                    (mapcar 'getvar a))
                 _vars))
   (princ)
)
;
; Restores the system variables in _VARS.
;
(defun POPVARS ()
   (if _vars (progn
      (mapcar '(lambda (x)
              (cond ((= (car x) "CECOLOR")
```

รูปที่ 3 (ต่อ) โปรแกรม ACAD. LSP

```
                        (setcolor (cadr x)))
                    ((= (car x) "CELTYPE")
                        (setltype (cadr x)))
                    (t (setvar (car x) (cadr x)))
                ))
                _vars)
            (setq _vars nil) )
        )
        (princ)
)
;
;******************** ERROR FUNCTION *************
;
;  Resets variables and Errors out.
;
(defun *ERROR* (st)
  (popvars)
  (terpri)
  (princ "\nError: ")
  (prompt (strcat st "\n"))
)
```

ตั้งโดยปกติตัวแรกจะเป็น A หรือกริดตามแนวนอนจะเริ่มจาก 1 จากนั้นจะเป็น a0 และ b0 ซึ่งจะถามจุดปลายทั้งสองของเส้นแนวเสาต้นแรก ส่วน d และ ang จะเป็นระยะและมุมระหว่างสองจุดนั้นตามลำดับ

ค่าของ a จะเป็นลิสต์ของจุดของกริดเสาต่าง ๆ ที่ได้จากฟังก์ชัน formlist ซึ่งจะต้องใส่เรียงลำดับไปเรื่อย ๆ เช่น กริดแนวนอนก็จะเรียงไปจนกว่าจะครบตามต้องการ กริดตั้งก็เช่นเดียวกัน ห้ามใส่สลับกัน จากนั้นก็เอามา cons รวมกับจุดที่เป็นตำแหน่งของกริดแรกคือ a0 สำหรับค่าของ b ก็เป็นไปทำนองเดียวกัน จากนั้นเราจะนำลิสต์ a มาแยกเป็นสองลิสต์ คือ a1 และ a2 ตามกรรมวิธีที่ได้อธิบายไปในไมโครฯเล่มที่ 47 โดยที่เราจะเอาสมาชิกในลิสต์มาทำอะไรกันได้ เราจะต้องแตกลิสต์นั้นเป็นสองลิสต์ก่อน ซึ่งการทำอย่างไรและเพื่ออะไรนั้นผมจะไม่ขอกล่าวซ้ำในที่นี้ (ขอได้โปรดกลับไปดูในไมโครเล่มที่ 47 เรื่องเดียวกันนี้) บรรทัดต่อไปจะดูว่าเส้นแนวเสานั้นเป็นแนวตั้งหรือแนวนอน ซึ่งโปรแกรมนี้จะเขียนชื่อของกริดเสาตามแบบสากลนิยม คือ ตามแนวนอนจะเป็นตัวอักษรเช่น A, B, C,... ส่วนในแนวตั้งจะเป็นตัวเลขเช่น 1, 2, 3, และจะเซตค่าสำหรับคำสั่ง dim เพื่อจะได้ทำได้ถูกต้องเป็น HOR(izontal) หรือ VER(tical) จากนั้นจะเป็นการสร้าง layer ขึ้นมาใหม่ โดยกำหนดให้กริดเสาและแนวระยะต่าง ๆ นี้จะถูกวาดขึ้นบน layer ที่มีชื่อว่า GRID โดยการใช้ฟังก์ชัน tblsearch ทดสอบดูว่ามี layer นี้อยู่แล้วหรือไม่ ซึ่งถ้ามีฟังก์ชันนี้จะให้ค่ามาเป็นลิสต์ของ definition ของ layer นี้

ชุดบรรทัดต่อมาจะเป็นหัวใจของโปรแกรม GR นี้ คือชุดของ mapcar และ lambda ซึ่งจะทำหน้าที่เปลี่ยนเส้นเป็นแบบ center และวาดเส้นแนวเสาใส่ bubble และเขียนบอกชื่อของกริดเสาอย่างถูกต้องตามสีและขนาดตัวหนังสือฯลฯ โดย mapcar และ lambda จะทำหน้าที่ของมันไปจนกว่าจะครบจำนวนสมาชิกในลิสต์ที่ส่งต่อให้กับฟังก์ชัน ในที่นี้ก็คือ a และ b บรรทัดต่อมาเป็นบรรทัดของการเขียนระยะโดยใช้คำสั่ง dim1 กับลิสต์ a1 และ a2 ส่วนที่เหลือก็เป็นการเซตค่าต่าง ๆ กลับคืนตามเดิม มีข้อแม้อีกนิดเดียวคือ ขอให้ใช้โปรแกรมนี้กับออโต้แคด เวอร์ชัน 10 เท่านั้น แต่ถ้าท่านประสงค์จะใช้กับออโต้แคดรุ่นเก่าก็สามารถที่จะทำได้โดยการแก้ไขตัวแปรระบบบางตัวที่ออโต้แคดรุ่นเก่าไม่มี โดยลบชื่อออกจากบรรทัด push-vars และ setqv

ขึ้นต้นเป็นออโต้แคด เวอร์ชัน 10 แต่กลายมาเป็นเรื่องนี้ได้ก็เพราะควันหลงจากงานสัมมนาวันนั้นแท้ ๆ เทียว อย่างไรก็ดี หวังใจว่าโปรแกรมที่ได้แสดงเป็นตัวอย่างในบทนี้คงจะเป็นประโยชน์แก่ท่านผู้อ่านไม่มากก็น้อย ในฉบับหน้าจะได้เข้าเรื่องออโต้แคด เวอร์ชัน 10 กันต่อครับ สวัสดี